UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/581,641	06/05/2006	Reinder Jaap Bril	US040013	5401

24737 7590 03/15/2010
PHILIPS INTELLECTUAL PROPERTY & STANDARDS
P.O. BOX 3001
BRIARCLIFF MANOR, NY 10510

EXAMINER	
CHEW, BRIAN	
ART UNIT	PAPER NUMBER
2195	
MAIL DATE	DELIVERY MODE
03/15/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/581,641	Applicant(s) BRIL ET AL.
	Examiner BRIAN CHEW	Art Unit 2195

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *6/5/2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *6/05/2006* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-26 are presented for examination.

Drawings

2. The drawings are objected to under 37 CFR 1.83(a) because they fail to show that control and logic module 451 comprises data receiver 452, evaluator 453, allocator 454, selector 455 and scheduler 501 (i.e. They appear to be six different entities in Figure 4B) as described in the specification. Any structural detail that is essential for a proper understanding of the disclosed invention should be shown in the drawing. MPEP § 608.02(d). Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as “amended.” If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either “Replacement Sheet” or “New Sheet” pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner,

the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

4. The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.

Arrangement of the Specification

As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.
(b) CROSS-REFERENCE TO RELATED APPLICATIONS.
(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT.
(d) THE NAMES OF THE PARTIES TO A JOINT RESEARCH AGREEMENT.
(e) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC.
(f) BACKGROUND OF THE INVENTION.
(1) Field of the Invention.
(2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.
(g) BRIEF SUMMARY OF THE INVENTION.
(h) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).
(i) DETAILED DESCRIPTION OF THE INVENTION.
(j) CLAIM OR CLAIMS (commencing on a separate sheet).
(k) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).
(l) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a

> nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

5. The disclosure is objected to because of the following informalities:

Page 2 line 1 recites references numbers [4] and [5], but it is uncertain what these reference numbers correspond to in the specification, and it is also uncertain whether these are intended to refer to an outside reference.

Page 2 line 31 recites reference number [6] with the same deficiencies.

Appropriate correction is required.

Claim Objections

6. Claims 1, 5 and 25-26 are objected to because of the following informalities:

Claim 1, line 8, recites, "processing one the tasks allocated to the processor". There appears to be a typographical error and should be corrected to read --processing one <u>of</u> the tasks allocated to the processor--.

Claim 5, lines 3-4, recites "in response to a task termination, repeating said step of identifying availability of memory in response to a task terminating". There appears to be a typographical error and should be corrected to read --in response to a task terminating, repeating said step of identifying availability of memory--.

Claim 25, line 2, recites "the method according to of claim 1". There appears to be a typographical error and should be corrected to read --the method according to claim 1--.

Claim 26, line 2, recites “the method according to of claim 1”. There appears to be a typographical error and should be corrected to read --the method according to claim 2--.

Appropriate correction is required.

7. Claim 14 is objected to because they include reference characters which are not enclosed within parentheses.

Reference characters corresponding to elements recited in the detailed description of the drawings and used in conjunction with the recitation of the same element or group of elements in the claims should be enclosed within parentheses so as to avoid confusion with other numbers or characters which may appear in the claims. See MPEP § 608.01(m).

8. Claims 25 and 26 objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form.

9. Claim 25 recites “a computer program, comprising a set of instructions arranged to cause a processing system to perform the method according to claim 1”. There are no additional method steps recited therein, and the nature of the computer program

does not alter any of the method steps of claim 1. As such, claim 25 fails to further limit claim 1.

10. Claim 26 recites "a computer program, comprising a set of instructions arranged to cause a processing system to perform the method according to claim 2". There are no additional method steps recited therein, and the nature of the computer program does not alter any of the method steps of claim 2. As such, claim 26 fails to further limit claim 2.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

11. Claims 14-16 and 25-26 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

12. As per claims 14-16, lines 14-15 of claim 14 recite “a selector arranged to select at least one task for suspension during execution of the task”. The specification lacks any description of a selector performing or being capable of performing this task. Specifically, the specification only discloses “a selector 455 for selecting tasks to initiate and terminate execution thereof” (specification, page 9, lines 31-32).

13. As per claims 25-26, there is no description in the specification of a computer program or instructions that cause a processing system to perform the methods of claims 1 or 2.

14. Claims 1-26 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claim language in the following claims is not clearly understood:

i. As per claim 1: Line 3, it is uncertain how the “providing” is performed (i.e. Provided by the task itself via interface Int_3? Providing to task manager 503?); Lines 3-4, it is uncertain what is meant by “suspension data specifying suspension of the task based on memory used thereby” (i.e. Suspend if these conditions are met? The conditions have been met so suspend now? What are the conditions? Memory used at a preemption point? Maximum amount of memory needed between preemption points?); Lines 7 and 9-10, it is uncertain what is meant by "each processor performing the steps of… monitoring an input indicative of memory used by the task matching the suspension data associated with the task" (i.e. Is this referring to a preemption point? Is the monitoring being performed by the processor? Or the evaluator?); Line 11, it is uncertain what is performing "suspending said task on the basis of said monitored input" (i.e. The claim states that this is performed by the processors but the specification states that this is performed by the scheduler 501).

ii. As per claim 2: Line 4, it is uncertain how tasks are allocated "having every task allocated to every processor" (i.e. The entire plurality of tasks assigned to each processor? Each task being able to be allocated to any processor?); Lines 5-6, it is uncertain how tasks are allocated "having every task allocated to a subset of said plurality of processors" (i.e. Each task allocated to each processor in the subset? Each task being able to be allocated to any processor in the subset?).

v. As per claim 4: Lines 2-4, it is uncertain what "first and second data" corresponds to (i.e. The suspension data? The monitored input? Neither?). Also, it is uncertain what "maximum memory usage" means (i.e. Threshold for allowable memory usage? The maximum amount of memory that was actually used?).

vi. As per claim 6: Line 2, it is uncertain what is meant by “the monitoring step is deemed unnecessary” (i.e. Is the monitoring step still performed? Is this referring to “monitoring termination of tasks” or “monitoring for an input”).

vii. As per claims 7 and 10, they have the same deficiencies as claim 4. Appropriate correction is required.

viii. As per claims 8 and 11, they have the same deficiencies as claim 5. Appropriate correction is required.

ix. As per claims 9 and 12, they have the same deficiencies as claim 6. Appropriate correction is required.

x. As per claim 13, it has the same deficiencies as claims 1 and 2.

xi. As per claim 14: It has the same deficiencies as claim 2; Line 1, it is uncertain whether “a scheduler” refers to (i.e. Is it referring to scheduler 501? Referring to control and logic modules 451? Referring to all elements of Figure 5? Isn't task manager 503 the data receiver?); Lines 7-8, it is uncertain how the evaluator can “identify, on the basis of the received data, whether there is sufficient memory to execute the tasks” (i.e. Data was only received for “a task” in lines 5-6 - where is the data for the rest of the tasks?); Line 21, it is uncertain what “its” refers to (i.e. The selector’s? The selected at least one task’s?); Line 23, it is uncertain what “the specified memory” refers to (i.e. Its specified memory usage? The specified amount of memory available to the data processing system?).

xii. As per claim 17: It has the same deficiencies as claim 2; Lines 6-7, it is uncertain how the evaluator can “identify, on the basis of the received data, whether there is sufficient memory to execute the tasks” (i.e. Data was only received for “a task” in lines 4-5 - where is the data for the rest of the tasks?); Lines 7-8, 14, and 16, it is uncertain whether “the tasks”, “said plurality of tasks” and “the plurality of tasks” refer to the same entity (i.e. A plurality of tasks from line 2); Lines 14-15, it is uncertain what “input received from the evaluating means” refers to (i.e. Isn’t the receiving means the only element that received anything? Is this referring to the identifying whether there is sufficient memory to execute the tasks?); Lines 18, it is uncertain what “the task” refers to (i.e. The “at least one task” from line 17? “The task" from line 5? “A task” from line 3?)

xiii. As per claim 18: It has the same deficiencies as claim 2; Lines 9-12, it is uncertain how “transmitting” is performed (i.e. Is this the same as “transmit deschedule instructions to the scheduler 501” on specification, page 6 line 26? If not, where is data coming from and going to?); Lines 11-12, it is uncertain what is meant by “specifying suspension of each task of said plurality of tasks based on memory usage during processing thereof" (i.e. Suspend if these conditions are met? The conditions have been met so suspend now?).

xiv. As per claim 19: Line 2, it is uncertain what "maximum memory usage" means (i.e. Threshold for allowable memory usage? The maximum amount of memory that was actually used?).

xv. As per claim 21: Lines 2-3, it is uncertain how the "at least one point… corresponds to each such sub-job" (i.e. One point for each sub-job? A point between each sub-job?).

xvi. As per claim 22, it has the same deficiencies as claim 19. Appropriate correction is required.

xvi. As per claim 23, it has the same deficiencies as claim 21. Appropriate correction is required.

xvii. As per claim 24: It has the same deficiencies as claims 2; Lines 2-3, it is uncertain what is meant by "associating suspension data with the task" (i.e. Is this the same as "providing suspension data" from claim 1? Otherwise, where is the "associating" described in the specification?); Line 3, "the task" lacks antecedent basis in the claim and will herein be construed as "a task"; Lines 3-4, it is uncertain what is meant by "specifying suspension of the task based on memory usage associated therewith" (i.e. Suspend if these conditions are met? The conditions have been met so suspend now?); Lines 7 and 13-14, it is uncertain what is meant by "each processor performing the steps of… monitoring an input indicative of memory used by the task matching the suspension data associated with the task" (i.e. Is this referring to a preemption point?).

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

15. Claims 25-26 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

16. As per claims 25-26, these claims are directed to a computer program, but lack necessary physical components (hardware) to constitute a machine or manufacture, such as a computer-readable medium for storing the computer program. Software does not fall within process, machine, manufacture or composition of matter categories of patentable subject matter recited in section 101.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 1-3, 13 and 18-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant-Admitted Prior Art (hereinafter AAPA) in view of Narlikar ("Space-Efficient Implementation of Nested Parallelism", 7/1997, ACM, pages 25-36).

18. As per claim 1, AAPA teaches a method of scheduling a plurality of tasks in a data processing system *("A known method of scheduling a plurality of tasks in a data processing system", specification, page 1, line 31)*, comprising the steps of:

- for each task of the plurality, providing suspension data specifying suspension of the task based on memory used thereby *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1)*; and each processor performing the steps of *("'task' is used to denote a unit of execution that can compete on its own for system resources such as memory, CPU", specification,* *page 1, lines 19-20; instructions of the method must be executed by a processor)*:

- monitoring for an input indicative of memory used by the task matching the suspension data associated with the task *("At run time… when it arrives at preemption points", specification, page 2, lines 18-19;"Data indicative of memory usage of a task conforming to the suspension data associated with each sub-job… requests a descheduling event, specifying that a preemption point has been reached", specification, page 2, lines 9-12)*; and
- suspending said task on the basis of said monitored input *("suspension of a task is referred to as task preemption", specification, page 1, lines 18-19; "preemption may occur at a preemption point", specification, page 2, line 21)*

but does not teach having a plurality of processors; allocating each of said plurality of tasks to a processor of said plurality of processors; processing one the tasks allocated to the processor; and processing a different one of the tasks allocated to the processor.

However, Narlikar teaches:

- having a plurality of processors *("The runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2)*;
- allocating each of said plurality of tasks to a processor of said plurality of processors *("while there exist threads in the system… remove-thread(out-queue)", Narlikar, figure 4, lines 2-3)*;

- processing one the tasks allocated to the processor *("execute the computation associated with τ", Narlikar, figure 4, line 6)*; and
- processing a different one of the tasks allocated to the processor *("remove-thread(out-queue)", Narlikar, figure 4, lines 2-3; "When a thread suspends, the processor accesses a new thread in a non-blocking manner from a work queue", Narlikar, section 1, paragraph 4, lines 13-14)*

"to minimize space usage of the parallel program" *(Narlikar, abstract, lines 8-9)*.

One of ordinary skill in the art at the time the invention was made would have been motivated to modify AAPA with the teachings of Narlikar because it "significantly reduces memory usage… without compromising performance" *(Narlikar, abstract, lines 17-18)*.

19. As per claim 2, AAPA in view of Narlikar teaches the method of claim 1, wherein allocation of a task to a processor is based on one of fixed allocation having every task allocated to a particular processor; variable allocation having every task allocated to every processor; and mixed allocation having every task allocated to a subset of said plurality of processors *("a thread may be moved from processor to processor at every timestep", Narlikar, section 1, paragraph 3, lines 20-21)*.

20. As per claim 3, AAPA in view of Narlikar teaches the method of claim 2, wherein said input comprises data indicative of a suspension request *("Data indicative of memory usage of a task conforming to the suspension data associated with each sub-*

job… requests a descheduling event, specifying that a preemption point has been reached", specification, page 2, lines 9-12).

21. As per claim 13, AAPA in view of Narlikar teaches a method of scheduling a plurality of tasks in a data processing system *("A known method of scheduling a plurality of tasks in a data processing system", specification, page 1, line 31)* having a plurality of processors *("The runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2)*, comprising the steps of:

- for each task of the plurality, providing suspension data specifying suspension of the task based on memory used thereby *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1)*;
- allocating each of said plurality of tasks to a processor of said plurality of processors based on one of-fixed allocation having every task allocated to a particular processor, variable allocation having every task allocated to every processor, and mixed allocation having every task allocated to a subset of said plurality of processors *("while there exist threads in the system… remove-thread(out-queue)", Narlikar, figure 4, lines 2-3) ("a thread may be moved from processor to processor at every timestep", Narlikar, section 1, paragraph 3, lines 20-21)*; and each processor performing the steps of:

- processing one the tasks allocated to the processor *("execute the computation associated with τ", Narlikar, figure 4, line 6)*;
- monitoring for an input indicative of memory used by the task matching the suspension data associated with the task *("At run time… when it arrives at preemption points", specification, page 2, lines 18-19;"Data indicative of memory usage of a task conforming to the suspension data associated with each sub-job… requests a descheduling event, specifying that a preemption point has been reached", specification, page 2, lines 9-12)*;
- suspending said task on the basis of said monitored input *("suspension of a task is referred to as task preemption", specification, page 1, lines 18-19; "preemption may occur at a preemption point", specification, page 2, line 21)*; and
- processing a different one of the tasks allocated to the processor *("remove-thread(out-queue)", Narlikar, figure 4, lines 2-3; "When a thread suspends, the processor accesses a new thread in a non-blocking manner from a work queue", Narlikar, section 1, paragraph 4, lines 13-14).*

22. As per claim 18, AAPA in view of Narlikar teaches a method of transmitting data to a data processing system *("A known method of scheduling a plurality of tasks in a data processing system", specification, page 1, line 31)* having a plurality of processors *("The runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2)*, the method comprising:

- allocating each task of a plurality of tasks to a processor of said plurality of processors based on one of fixed allocation having every task allocated to a particular processor, variable allocation having every task allocated to every processor, and mixed allocation having every task allocated to a subset of said plurality of processors *("while there exist threads in the system… remove-thread(out-queue)", Narlikar, figure 4, lines 2-3) ("a thread may be moved from processor to processor at every timestep", Narlikar, section 1, paragraph 3, lines 20-21)*;
- transmitting data for use by the data processing system in processing each task of said plurality of tasks *("they take threads off the out-queue", Narlikar, section 4, paragraph 4, lines 2-3)*;
- transmitting suspension data specifying suspension of each task of said plurality of tasks based on memory usage during processing thereof *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1) ("they take threads off the out-queue", Narlikar, section 4, paragraph 4, lines 2-3)*; wherein, the data processing system is configured to perform a process comprising:
- monitoring for an input indicative of memory usage of each task matching the suspension data associated with the task *("At run time… when it arrives at preemption points", specification, page 2, lines 18-19;"Data indicative of memory usage of a task conforming to the suspension data associated with each sub-*

job… requests a descheduling event, specifying that a preemption point has been reached", specification, page 2, lines 9-12); and

- suspending processing of said each task on the basis of said monitored input *("suspension of a task is referred to as task preemption", specification, page 1, lines 18-19; "preemption may occur at a preemption point", specification, page 2, line 21).*

23. As per claim 19, AAPA in view of Narlikar teaches the method of claim 18, wherein the suspension data includes data identifying maximum memory usage associated with said each task *("preemption points preferably coincide with the sub-job boundaries of the task", specification, page 2, lines 7-8; "preemption points can be utilized to avoid data processing system crashes due to a lack of memory", specification, page 2, lines 5-6; this describes the upper boundary, or maximum memory usage, associated with each task).*

24. As per claim 20, AAPA in view of Narlikar teaches the method of claim 18, wherein the suspension data identifies at least one point at which processing of each task can be suspended, based on memory usage of said each task *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1).*

25. As per claim 21, AAPA in view of Narlikar teaches the method of claim 20, Narlikar teaches the method of claim 20, wherein said each task comprises a plurality of sub-jobs and said data identifying at least one point at which processing of said each task can be suspended corresponds to each such sub-job *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1).*

26. As per claim 22, AAPA in view of Narlikar teaches the method of claim 20, wherein the suspension data includes data identifying maximum memory usage associated with said each task *("preemption points preferably coincide with the sub-job boundaries of the task", specification, page 2, lines 7-8; "preemption points can be utilized to avoid data processing system crashes due to a lack of memory", specification, page 2, lines 5-6; this describes the upper boundary, or maximum memory usage, associated with each task).*

27. As per claim 23, AAPA in view of Narlikar teaches the method of claim 22, wherein said each task comprises a plurality of sub-jobs and said data identifying at least one point at which processing of said each task can be suspended corresponds to each such sub-job *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for*

suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1).

28. As per claim 24, AAPA in view of Narlikar teaches a method of configuring a plurality of tasks for use in a data processing system *("A known method of scheduling a plurality of tasks in a data processing system", specification, page 1, line 31)* having a plurality of processors *("The runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2)*, the method including:

- associating suspension data with the task, the suspension data specifying suspension of the task based on memory usage associated therewith *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1)*; wherein the data processing system is arranged to perform a process in respect of a plurality of tasks executing on a plurality of processors, the process comprising:
- allocating each task of said plurality of tasks to a processor of said plurality of processors based on one of fixed allocation having every task allocated to a particular processor, variable allocation having every task allocated to every processor, and mixed allocation having every task allocated to a subset of said plurality of processors *("while there exist threads in the system… remove-*

thread(out-queue)", Narlikar, figure 4, lines 2-3) ("a thread may be moved from processor to processor at every timestep", Narlikar, section 1, paragraph 3, lines 20-21);

- monitoring for an input indicative of memory usage of the task matching the suspension data associated with the task *("At run time… when it arrives at preemption points", specification, page 2, lines 18-19;"Data indicative of memory usage of a task conforming to the suspension data associated with each sub-job… requests a descheduling event, specifying that a preemption point has been reached", specification, page 2, lines 9-12)*; and
- suspending processing of said task on the basis of said monitored input *("suspension of a task is referred to as task preemption", specification, page 1, lines 18-19; "preemption may occur at a preemption point", specification, page 2, line 21).*

29. As per claim 25, AAPA in view of Narlikar teaches a computer program, comprising a set of instructions arranged to cause a processing system to perform the method according to of claim 1 *("The runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2).*

30. As per claim 26, AAPA in view of Narlikar teaches a computer program, comprising a set of instructions arranged to cause a processing system to perform the

method according to of claim 2 *("The runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2).*

31. Claims 4-12 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over hereinafter AAPA in view of Narlikar, further in view of Bishop *et al.* (US 5,826,082; hereinafter Bishop).

32. As per claim 4, AAPA in view of Narlikar teaches the method of claim 3 discussed above, but does not teach further comprising the steps of: receiving first data identifying maximum memory usage associated with the plurality of tasks; receiving second data identifying memory available for processing the plurality of tasks; identifying, on the basis of the first and second data, whether there is sufficient memory available to process the tasks; and wherein, said monitoring and suspending steps are performed only in response to identifying insufficient memory.

However, Bishop teaches further comprising the steps of:

- receiving first data identifying maximum memory usage associated with the plurality of tasks *("In block 202, a thread submits a request to the resource manager to obtain the resources need to complete execution of an operation associated with the thread", Bishop, column 3, lines 55-57)*;
- receiving second data identifying memory available for processing the plurality of tasks *("block 204, the resource manager reads the available indicator of the*

Resource Status Table 125 to determine of the requested amount of the requested resource is available", Bishop, column 3, lines 64-67);

- identifying, on the basis of the first and second data, whether there is sufficient memory available to process the tasks *("block 204, the resource manager reads the available indicator of the Resource Status Table 125 to determine of the requested amount of the requested resource is available", Bishop, column 3, lines 64-67)*

for "reserving and managing computer system resources… set-top boxes… where resources (e.g. memory) are limited" *(Bishop, column 1, lines 5-11)*.

One of ordinary skill in the art at the time the invention was made would have been motivated to modify AAPA in view of Narlikar with the teachings of Bishop to "accommodate the uncertainty of whether resources needed to complete an operation will be available" *(Bishop, column 1, lines 13-14)*. The combination does not teach wherein said monitoring and suspending steps are performed only in response to identifying insufficient memory

However, it would've been obvious to one of ordinary skill in the art at the time the invention was made for wherein said monitoring and suspending steps are performed only in response to identifying insufficient memory. It would have been obvious not to perform the suspending step because there is sufficient memory for the task. As such, it would also have been obvious not to perform the monitoring step because doing so would waste resources monitoring for a suspension event that has already been determined not to occur.

33. As per claim 5, AAPA in view of Narlikar and Bishop teaches the method of claim 4, further comprising the steps of:

- monitoring termination of tasks *("after the third thread is finished with the memory", Bishop, column 5, lines 64-65)*; and
- in response to a task termination, repeating said step of identifying availability of memory *("after the third thread is finished with the memory, the resource manager will… unsuspend the second thread's request for memory", Bishop, column 5, lines 64-67; once the first thread finishes, there is sufficient memory to unsuspend the second request/thread)*.

34. As per claim 6, AAPA in view of Narlikar and Bishop teaches the method of claim 5, but does not teach in response to identifying sufficient memory to execute the remaining tasks, the monitoring step is deemed unnecessary.

However, it would've been obvious to one of ordinary skill in the art at the time the invention was made for in response to identifying sufficient memory to execute the remaining tasks, the monitoring step is deemed unnecessary. It would have been obvious that the monitoring step be deemed unnecessary because performing the step would waste resources monitoring for a suspension event that has already been determined not to occur, since there is sufficient memory for the task.

35. As per claim 7, AAPA in view of Narlikar and Bishop teaches the method of claim 6, and claim 7 is rejected for the same reason as claim 4 above.

36. As per claim 8, AAPA in view of Narlikar and Bishop teaches the method of claim 7, and claim 8 is rejected for the same reason as claim 5 above.

37. As per claim 9, AAPA in view of Narlikar and Bishop teaches the method of claim 8, and claim 9 is rejected for the same reason as claim 6 above.

38. As per claim 10, AAPA in view of Narlikar and Bishop teaches the method of claim 2, and claim 10 is rejected for the same reason as claim 4 above.

39. As per claim 11, AAPA in view of Narlikar and Bishop teaches the method of claim 10, and claim 11 is rejected for the same reason as claim 5 above.

40. As per claim 12, AAPA in view of Narlikar and Bishop teaches the method of claim 11, and claim 12 is rejected for the same reason as claim 6 above.

41. As per claim 17, AAPA in view of Narlikar and Bishop teaches a data processing system having a plurality of processors arranged to execute a plurality of tasks *("runtime system that schedules parallel threads", Narlikar, section 7, paragraph 1, line 9)("The*

runtime system has been implemented on a 16-processor SGI Power Challenge", Narlikar, section 6.1, paragraph 1, lines 1-2), the data processing system including:

- memory arranged to hold instructions and data during execution of a task *("The runtime system has… a shared memory architecture with processors and memory connected", Narlikar, section 6.1, paragraph 1, lines 1-3)*;
- receiving means arranged to receive data identifying maximum memory usage associated with a task *("In block 202, a thread submits a request to the resource manager to obtain the resources need to complete execution of an operation associated with the thread", Bishop, column 3, lines 55-57)* and data specifying preemptability of the task *("each sub-job of a task have a set of… suspension data, that specifies the processing preemption points and corresponding conditions for suspension of a sub-job based on its memory usage", specification, page 1 line 31 - page 2 line 1*;
- evaluating means arranged to identify, on the basis of the received data, whether there is sufficient memory to execute the tasks *("block 204, the resource manager reads the available indicator of the Resource Status Table 125 to determine of the requested amount of the requested resource is available", Bishop, column 3, lines 64-67)*;
- an allocator arranged to allocate each of said plurality of tasks to a processor of said plurality of processors based on one of fixed allocation having every task allocated to a particular processor, variable allocation having every task allocated to every processor, and mixed allocation having every task allocated to a subset

of said plurality of processors *("while there exist threads in the system… remove-thread(out-queue)", Narlikar, figure 4, lines 2-3) ("a thread may be moved from processor to processor at every timestep", Narlikar, section 1, paragraph 3, lines 20-21)*; and

- a scheduler arranged to schedule execution of the tasks *("execute the computation associated with τ", Narlikar, figure 4, line 6)* on the basis of input received from the evaluating means *("if the requested amount of the requested resource is not available, the requesting thread may choose not to begin its operation", Bishop, column 4, lines 21-23),*
- wherein, in response to identification of insufficient memory to execute the plurality of tasks, the scheduler is arranged to suspend execution of at least one task in dependence on memory usage by the task *("resource manager has found a prior request which has reserved enough of the requested resource to satisfy the present request… If the requested resource is unallocated to the prior request"; Bishop, column 5, lines 5-11; "resource manager will temporarily suspend the prior request", Bishop, column 5, line 28; "if the requested amount of the requested resource is not available, the requesting thread may choose not to begin its operation", Bishop, column 4, lines 21-23; since the resource request for the prior request is suspended, the thread corresponding to the prior request is suspended because resources are unavailable).*

42. Claims 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over Narlikar in view of Bishop.

43. As per claim 14, Narlikar teaches a scheduler for use in a data processing system having a plurality of processors, the data processing system being arranged to execute a plurality of tasks on said plurality of processors and having access to a specified amount of memory for use in executing the tasks *("runtime system that schedules parallel threads", Narlikar, section 7, paragraph 1, line 9)("The runtime system has been implemented on a 16-processor SGI Power Challenge, which has a shared memory architecture with processors and memory connected", Narlikar, section 6.1, paragraph 1, lines 1-3),* the scheduler comprising:

- an allocator arranged to allocate each of said plurality of tasks to a processor of said plurality of processors based on one of a fixed allocation having every task allocated to a particular processor, variable allocation having every task allocated to every processor, and mixed allocation having every task allocated to a subset of said plurality of processors *("while there exist threads in the system… remove-thread(out-queue)", Narlikar, figure 4, lines 2-3) ("a thread may be moved from processor to processor at every timestep", Narlikar, section 1, paragraph 3, lines 20-21)*; and
- a scheduler arranged to initiate execution of said allocated task and suspend said selected task *("execute the computation associated with τ… τ suspends", Narlikar, figure 4, lines 6-7)*

but does not teach a data receiver arranged to receive data identifying maximum memory usage associated with a task; an evaluator arranged to identify, on the basis of the received data, whether there is sufficient memory to execute the tasks; and a selector arranged to select at least one task for suspension during execution of the task, said suspension coinciding with a specified memory usage by the task; wherein, in response to the evaluator identifying that there is insufficient memory to execute the plurality of tasks, the selector selects at least one task for suspension, on the basis of its specified memory usage, and the specified amount of memory available to the data processing system, and the scheduler suspends execution of the at least one selected task in response to the task using the specified memory.

However, Bishop teaches:

- a data receiver arranged to receive data identifying maximum memory usage associated with a task *("In block 202, a thread submits a request to the resource manager to obtain the resources need to complete execution of an operation associated with the thread", Bishop, column 3, lines 55-57)*;
- an evaluator arranged to identify, on the basis of the received data, whether there is sufficient memory to execute the tasks *("block 204, the resource manager reads the available indicator of the Resource Status Table 125 to determine of the requested amount of the requested resource is available", Bishop, column 3, lines 64-67)*;
- a selector arranged to select at least one task for suspension during execution of the task, said suspension coinciding with a specified memory usage by the task

("resource manager has found a prior request which has reserved enough of the requested resource to satisfy the present request… If the requested resource is unallocated to the prior request"; Bishop, column 5, lines 5-11; "resource manager will temporarily suspend the prior request", Bishop, column 5, line 28; "if the requested amount of the requested resource is not available, the requesting thread may choose not to begin its operation", Bishop, column 4, lines 21-23; since the resource request for the prior request is suspended, the thread corresponding to the prior request is suspended because resources are unavailable);

- wherein, in response to the evaluator identifying that there is insufficient memory to execute the plurality of tasks, the selector selects at least one task for suspension, on the basis of its specified memory usage, and the specified amount of memory available to the data processing system, and the scheduler suspends execution of the at least one selected task in response to the task using the specified memory *("resource manager has found a prior request which has reserved enough of the requested resource to satisfy the present request… If the requested resource is unallocated to the prior request"; Bishop, column 5, lines 5-11; "resource manager will temporarily suspend the prior request", Bishop, column 5, line 28; "if the requested amount of the requested resource is not available, the requesting thread may choose not to begin its operation", Bishop, column 4, lines 21-23; since the resource request for the prior request is* *suspended, the thread corresponding to the prior request is suspended because resources are unavailable)*

for "reserving and managing computer system resources... set-top boxes... where resources (e.g. memory) are limited" *(Bishop, column 1, lines 5-11).*

One of ordinary skill in the art at the time the invention was made would have been motivated to modify Narlikar with the teachings of Bishop to "accommodate the uncertainty of whether resources needed to complete an operation will be available" *(Bishop, column 1, lines 13-14).*

44. As per claim 15, Narlikar and Bishop teaches the scheduler of claim 14, wherein the evaluator is arranged to monitor termination of tasks, and in response to a task terminating, to identify whether there is sufficient memory to execute the remaining tasks *("after the third thread is finished with the memory, the resource manager will... unsuspend the second thread's request for memory", Bishop, column 5, lines 64-67; once the first thread finishes, there is sufficient memory to unsuspend the second request/thread).*

45. As per claim 16, Narlikar and Bishop teaches the scheduler of claim 15, wherein, in response to the evaluator identifying sufficient memory to execute the remaining tasks, the selector is arranged to deselect said selected at least one task *("after the third thread is finished with the memory, the resource manager will... unsuspend the second*

thread's request for memory", Bishop, column 5, lines 64-67; once the first thread finishes, there is sufficient memory to unsuspend the second request/thread)).

Conclusion

46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to BRIAN CHEW whose telephone number is (571)270-5571. The examiner can normally be reached on Monday-Thursday, 8:00AM-5:00PM EST.

47. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng An can be reached on (571)272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/

Supervisory Patent Examiner, Art Unit 2195

/B.C./
Examiner
Art Unit 2195